Docket No. GB920020006US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Hindle et al.**	§	Group Art Unit: **2156**
	§	
Serial No. **10/537,213**	§	Examiner: **Al Hashemi, Sana A.**
	§	
Filed: **June 2, 2005**	§	Customer No.: **50170**
	§	
For: **Synchronizing Data in a**	§	
Distributed Data Processing System	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANTS' BRIEF (37 C.F.R. § 41.37)

This Appeal Brief is in furtherance of the Notice of Appeal filed May 18, 2010 (37 C.F.R. § 41.31).

The fees required under § 41.20(b)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying Fee Transmittal.

I. Real Party in Interest

The real party in interest in this appeal is the following party: International Business Machines Corporation.

II. Related Appeals and Interferences

With respect to other appeals and interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

III. Status of Claims

The status of the claims involved in this proceeding is as follows:

1. Claims canceled: 6-23
2. Claims withdrawing from consideration but not canceled: 24-31
3. Claims pending: 1-5 and 24-31
4. Claims allowed: NONE
5. Claims rejected: 1-5 and 24-31

The claims on appeal are: claims 1-5 and 24-31

IV. Status of Amendments

No amendments to the application were filed subsequent to mailing of the Final Office Action.

V. Summary of Claimed Subject Matter

Independent Claim 1:

A method of synchronizing data in a distributed data processing system comprises storing a master data (see 330 in FIG. 3, for example) in at least one legacy computer system (see 325 in FIG. 3; page 4, lines 1-14; page 8, lines 5-18; page 10, line 37, to page 11, line 9, for example). The method further comprises enabling a first non-legacy computer (see 315 in FIG. 3, for example) to support synchronization (see page 10, lines 32-38, for example). The method further comprises storing a copy (see 310 in FIG. 3, for example) of the master data in a second non-legacy computer (see 325 in FIG. 3; page 8, lines 20-25, for example). The method further comprises executing, by the second non-legacy computer, at least one operation on said copy of the master data (see 400 in FIG. 4; 500 in FIG. 5; page 8, lines 25-31, for example). The method further comprises sending, by said second non-legacy computer, the at least one operation to the first non-legacy computer (see 405-420 in FIG. 4; 505-515 in FIG. 5; page 9, lines 1-29, for example). The method further comprises replaying, by the first non-legacy computer, the at least one operation (see 520 in FIG. 5; page 9, lines 29-35, for example). The method further comprises determining whether the at least one operation is successful (see 425 in FIG. 4; page 9, line 36, to page 10, line 2, for example). In response to a determination that the at least one operation is successful, the method comprises synchronizing the master data by applying the at least one operation to form a modified copy of the master data (see 430 in FIG. 4; page 10, lines 2-9, for example).

Independent Claim 24:

An apparatus in a middle-tier computer comprises a processor and a memory coupled to the processor (see 315; page 3, lines 6-41; page 8, lines 5-18, for example). The memory comprises instructions which, when executed by the processor, cause the processor to receive, via a first software connector (see 331 in FIG. 3; page 8, lines 5-12, for example), at least one operation from a thin client computer (see 305 in FIG. 3; 405-420 in FIG. 4; 505-515 in FIG. 5; page 9, lines 1-29, for example). The thin client computer stores a copy (see 310 in FIG. 3, for example) of master data (see 330 in FIG. 3, for example) from a legacy computer (see 325 in FIG. 3; page 4, lines 1-14; page 8, lines 5-18; page 10, line 37, to page 11, line 9, for example)

and executes the at least one operation on the copy of the master data (see 400 in FIG. 4; 500 in FIG. 5; page 8, lines 25-31, for example). The instructions further cause the processor to sequentially replay the at least one operation on the master data at the legacy computer via a second software connector (see 331 in FIG. 3; 520 in FIG. 5; page 8, lines 5-12; page 9, lines 29-35, for example). The instructions further cause the processor to determine whether the at least one operation is successful (see 425 in FIG. 4; page 9, line 36, to page 10, line 2, for example). In response to a determination that the at least one operation is successful, the instructions cause the processor to synchronize the master data by applying the at least one operation via the second software connector to form new master data at the legacy computer (see 430 in FIG. 4; page 10, lines 2-9, for example), such that in response to a determination that the at least one operation is not successful, the middle-tier computer does not synchronize the master data (see 435 in FIG. 4; page 10, lines 2-9, for example).

***Independent Claim 28*:**

A computer program product comprises a computer recordable medium having a computer readable program recorded thereon (see page 7, lines 16-18, for example). The computer readable program, when executed on a middle tier computer, causes the middle tier computer to receive, via a first software connector (see 331 in FIG. 3; page 8, lines 5-12, for example), at least one operation from a thin client computer (see 305 in FIG. 3; 405-420 in FIG. 4; 505-515 in FIG. 5; page 9, lines 1-29, for example). The thin client computer stores a copy (see 310 in FIG. 3, for example) of master data (see 330 in FIG. 3, for example) from a legacy computer (see 325 in FIG. 3; page 4, lines 1-14; page 8, lines 5-18; page 10, line 37, to page 11, line 9, for example) and executes the at least one operation on the copy of the master data (see 400 in FIG. 4; 500 in FIG. 5; page 8, lines 25-31, for example). The computer readable program further causes the middle tier computer to sequentially replay the at least one operation on the master data at the legacy computer via a second software connector (see 331 in FIG. 3; 520 in FIG. 5; page 8, lines 5-12; page 9, lines 29-35, for example). The computer readable program further causes the middle tier computer to determine whether the at least one operation is successful (see 425 in FIG. 4; page 9, line 36, to page 10, line 2, for example). In response to a determination that the at least one operation is successful, the computer readable program causes

the middle tier computer to synchronize the master data by applying the at least one operation via the second software connector to form new master data at the legacy computer (see 430 in FIG. 4; page 10, lines 2-9, for example), such that in response to a determination that the at least one operation is not successful, the middle-tier computer does not synchronize the master data (see 435 in FIG. 4; page 10, lines 2-9, for example).

VI. Grounds of Rejection to be Reviewed on Appeal

A. The Office rejects at least claim 1 under 35 U.S.C. § 112, second paragraph, as allegedly failing to particularly point out and distinctly claim the subject matter that Appellants regard as the invention. (The Final Office Action does not include a statement of rejection; therefore, it is unclear what claims are rejected, if any.)

B. The Office rejects claims 1-5 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Gehman et al. (U.S. Patent No. 7,136,881) in view of Grimsrud (U.S. Patent No. 6,546,437).

C. The Office rejects claims 24-31 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Gehman et al. (U.S. Patent No. 7,136,881) in view of Grimsrud (U.S. Patent No. 6,546,437) and Yousefi'zadeh (U.S. Patent No. 6,950,848). (The Final Office Action states that claims 1-5 are rejected; however, the body of the rejection addresses claims 24-31.)

VII. Argument

A. 35 U.S.C. § 112, Alleged Indefiniteness of Claims 1-5

The Office rejects at least claim 1 under 35 U.S.C. § 112, second paragraph, as allegedly failing to particularly point out and distinctly claim the subject matter that Applicants regard as the invention. (The Final Office Action does not include a statement of rejection; therefore, it is unclear what claims are rejected, if any.) Appellants respectfully traverse this rejection.

With respect to claim 1, the Final Office Action states that it is unclear to the Examiner how the non-legacy computer executes an operation on a master version of data stored on a legacy computer, since the body of the claim does not clearly show how the legacy and non-legacy computers communicate. However, as argued in the Response to Final Office Action

filed August 11, 2009, it is not the role of the claims to teach one skilled in the art to reproduce the invention, but rather to define the legal metes and bounds of the invention. *In re Rainer*, 305 F.2d 505, 509, 134 U.S.P.Q. 343, 346 (C.C.P.A. 1962). If the metes and bounds of the claimed invention are clearly ascertainable, then the claim cannot be properly rejected as "vague" or "indefinite" under 35 U.S.C. § 112, second paragraph. In this case, the scope of claim 1 is clear, even if the manner of implementation is unclear to the Examiner. Whether the claim leaves unclear the manner in which the feature of a non-legacy computer executing an operation on a master version of data stored on a legacy computer may be implemented is irrelevant where the claim clearly covers all forms of implementation. *In re Warmerdam*, 33 F.3d 1354, 1361, 31 U.S.P.Q.2d 1754, 1759 (Fed. Cir. 1994).

Throughout the prosecution of this application, the Examiner has not addressed this argument. Therefore, Appellants respectfully request that the rejection of claim 1 under 35 U.S.C. § 112, second paragraph, not be sustained.

B. <u>35 U.S.C. § 103, Alleged Obviousness of Claims 1-5</u>

The Office rejects claims 1-5 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Gehman et al. (U.S. Patent No. 7,136,881) in view of Grimsrud (U.S. Patent No. 6,546,437). Appellants respectfully traverse this rejection.

Gehman appears to teach a method and system for processing directory events. An event master server 40 records modification messages in a queue 40b. Event master server 40 includes event message provider 40a, which sends the modification messages as event messages to event service server 41, which includes replicate data monitor 41a. Event service server 41 stores sequence numbers for event messages in queue 41b. Replicate data monitor 41a sends event messages to event client server, which includes event notifier 42a and directory client register 42b. In this way, changes to a master directory database may be replicated or synchronized to directory clients. That is, changes at a master directory database are perpetuated down to clients.

In contradistinction, the claim 1 recites a method of synchronizing data in a distributed data processing system. The method stores a master data in at least one legacy computer system and enables a first non-legacy computer to support synchronization. A second non-legacy computer (e.g., a client) stores a copy of the master data in a second non-legacy computer, executes at least one operation on the copy of the master data, and sends the at least one

operation to the first non-legacy computer. The first non-legacy computer executes the at least one operation on the master data at the at least one legacy computer. The method determines whether the first non-legacy computer successfully executed the at least one operation and synchronizes the master data by applying the at least one operation in response to the first non-legacy computer successfully executed the at least one operation.

Gehman does not teach or suggest executing at least one operation on a copy of the master data in a second non-legacy computer. Rather, Gehman teaches perpetuating changes **from** the master directory database **to** the clients, not the other way around. Therefore, it follows that Gehman also fails to teach sending, by the second non-legacy computer, the at least one operation to the first non-legacy computer and executing, by the first non-legacy computer, the at least one operation on the master data at the at least one legacy computer, because Gehman teaches an event service server that perpetuates changes from the master directory database to clients and fails to teach a non-legacy computer that receives changes made at the client and executes those changes on a legacy computer.

In response to the above argument, the Final Office Action issued May 15, 2009, states that Gehman teaches manipulating data within master directory database and replicating the manipulated data within master directory event system to a client corresponding to the copy of master data to a non-legacy computer at col. 3, lines 40-55. In the Response to Final Office Action filed August 11, 2009, Appellants noted that this portion of Gehman reinforces Appellants' argument, because Gehman clearly teaches that only the master data is manipulated at the legacy computer, that manipulations made at the legacy computer are replicated to the client computer, and that no manipulations are made on the replicated data at the client computer. Therefore, not only does Gehman not teach executing at least one operation on a copy of the master data in a second non-legacy computer, but there is no need to synchronize such manipulations back up to the legacy computer.

The Final Office Action issued March 10, 2010, provides no rebuttal for the above argument other than to state that the limitation is rejected under 35 U.S.C. § 112, second paragraph, for being unclear. Appellants maintain that the scope of sending, by the second non-legacy computer, the at least one operation to the first non-legacy computer and executing, by the first non-legacy computer, the at least one operation on the master data at the at least one legacy

computer can be clearly determined, especially when read in light of the specification. The rejection under 35 U.S.C. § 112, second paragraph, asks how things are done, which is a question that is easily answered within the specification. The rejection under 35 U.S.C. § 112, second paragraph, does not raise a question concerning whether the scope of the claim is vague or indefinite. Regardless, claim 1 is clear on its face and presents features that are not taught or suggested by the cited prior art. The Final Office Action fails to establish that the features are taught or suggested by Gehman; therefore, the Final Office Action fails to establish a *prima facie* case of obviousness.

Furthermore, Gehman does not teach or suggest a determination of whether the first non-legacy computer successfully executed the at least one operation. The Final Office Action issued May 15, 2009, alleged that Gehman teaches such a determination because Gehman teaches an option of "yes" in step S96 in FIG. 3B, which is reproduced as follows:

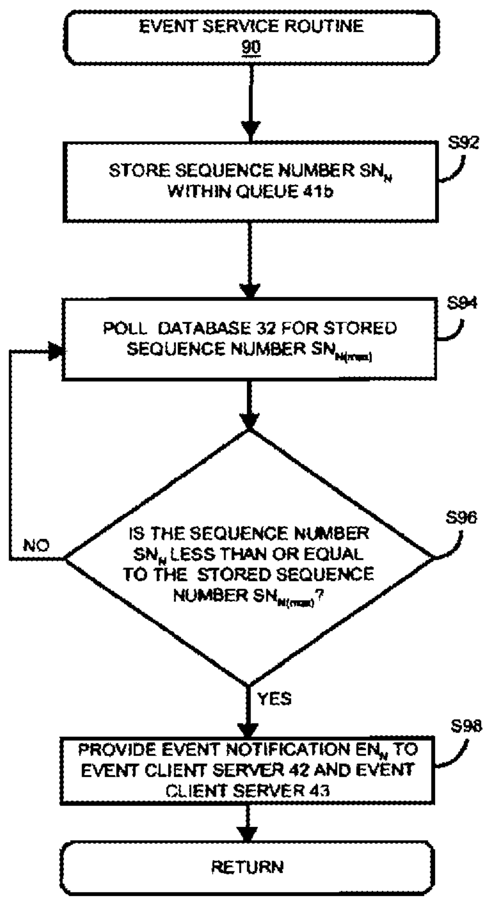


FIG. 3B

As seen in FIG. 3B of Gehman, step S96 determines whether a sequence number is less than or equal to the stored sequence number. This determination decides whether to send an event notification to an event client server. Gehman states:

> Referring additionally to FIG. 3B, during stage S92 of routine 90, replicate data monitor 41a stores sequence number SN_N within queue 41b. Replicate data monitor 41a proceeds to stage S94 of routine 90 to poll replicate directory database 32 for a sequence number SN_{MAX} stored therein. Those of ordinary skill in the art will appreciate there can be a significant gap of time from a completion of the writing of sequence number SN_N to master directory database 31 by event message provider 40a and a subsequent completion of the replication of all of the data within master directory database 31, including the manipulated data and sequence number SN_N, to replicate directory database 32. As such, those of ordinary skill in the art will further appreciate that sequence number SN_N being greater than sequence number SN_{MAX} is an indication that the replication of all of the data within master directory database 31, including the manipulated data and sequence number SN_N, has not occurred. Conversely, those of ordinary skill in the art will further appreciate that sequence number SN_N being less than or equal to sequence number SN_{MAX} is an indication that the replication of all of the data within master directory database 31, including the manipulated data and sequence number SN_N, has occurred.
>
> Thus, during stage S96 of routine 90, replicate data monitor 41a compares sequence number SN_N and sequence number SN_{MAX} (to determine if sequence number SN_N is less than or equal to sequence number SN_{MAX}. If sequence number SN_N is greater than sequence number SN_{MAX}, replicate data monitor 41a loops back to stage S94. If sequence number SN_N is less than or equal to sequence number SN_{MAX}, replicate data monitor 41a proceeds to stage S98 to provide event notification EN_N (or an edited version thereof) to event client server 42 and event client server 43.

Gehman, col. 4, line 39, to col. 5, line 3. Therefore, Gehman teaches a determination concerning the sequence of event notifications, but does not teach a determination of whether a first non-legacy computer successfully executes at least one operation on a legacy computer, wherein the at least one operation is received from a second non-legacy computer.

In response to the above argument, the Final Office Action issued May 15, 2009, stated that Gehman teaches such a determination in a "YES" and "NO" in step S82 of FIG. 3A, which is as follows:

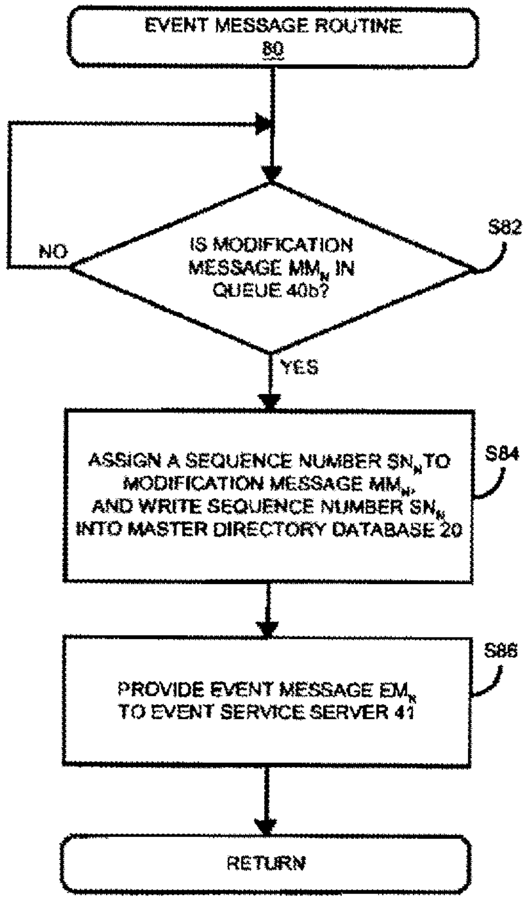
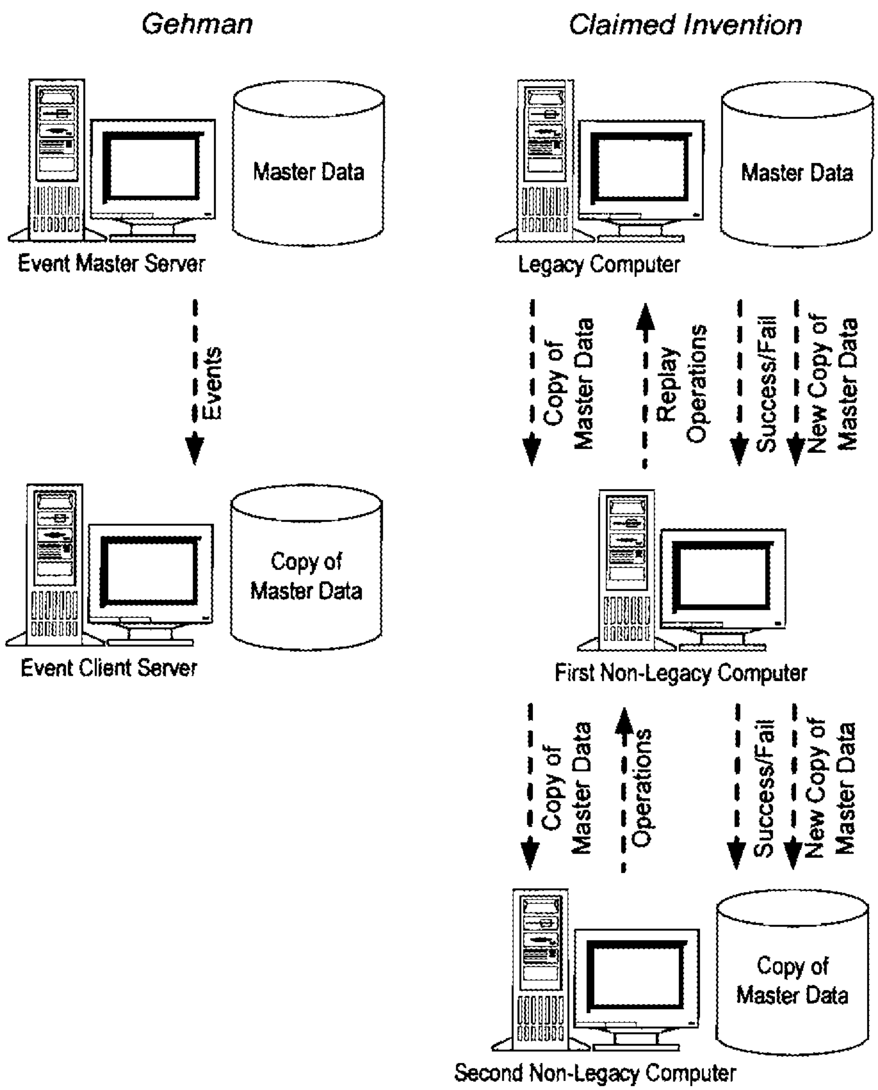


FIG. 3A

As argued in the Response to Final Office Action filed August 11, 2009, clearly, in step S82, Gehman teaches a determination as to whether a modification message exists in a queue to be applied to the master data at the legacy computer. Gehman does not teach that such a modification message corresponds to execution of operation on a **copy** of master data at a non-legacy computer. Furthermore, the Final Office Action proffers no technical analysis or explanation as to why existence of a modification message in a queue is somehow equivalent to a determination of whether at least one operation received from a non-legacy computer that was executed on a copy of master data was successfully replayed on master data at a legacy computer.

The Final Office Action issued March 10, 2010, does not provide a rebuttal to the above argument.

As shown in FIG. 2 of Gehman, changes at the event master server are perpetuated to the event client server using event messages. The distinctions between Gehman and the presently claimed invention are illustrated as follows:



According to claims 1-5, the method of synchronization provides a way for the second non-legacy computer to perform operations on a local copy of the master data and **also** to have those operations replayed by a first non-legacy computer. While Gehman may be similar in that changes to master data are perpetuated to other computers, Gehman does not teach the **specific combination of features** recited in claims 1-5.

Still further, the Final Office Action acknowledges that Gehman does not teach that one computer is a legacy computer and other computers are non-legacy computers. The Final Office

Action alleges that Grimsrud generally teaches both legacy and non-legacy computers and concludes that it would have been obvious to a person of ordinary skill in the art to incorporate legacy and non-legacy computers in Gehman. Appellants respectfully disagree.

Grimsrud teaches communication between a computer and a peripheral with a legacy failure control mechanism. Grimsrud teaches a computer 174 connected to a peripheral 176 via an advanced technology attachment (ATA) interface 178. The computer 174 may send a request for information to the peripheral, and the peripheral may send a reply to the computer that causes the computer to not use the peripheral. See Grimsrud, col. 1, lines 30-56. The computer and peripheral can be any combination of legacy and non-legacy in various embodiments of Grimsrud.

However, as argued in the Response to Final Office Action filed August 11, 2009, Grimsrud does not teach a person of ordinary skill in the art to modify Gehman to include legacy and non-legacy computers. There is no problem in Gehman for which Grimsrud is a solution. The Final Office Action proposes that a person of ordinary skill in the art would have been motivated to combine the teachings of Gehman and Grimsrud "in order to the method of detecting any changes or any hardware connected to the system." While this may be a problem solved in Grimsrud, this motivation does not apply to Gehman in any significant way.

In response to the above argument, the Final Office Action repeats the motivation of improving performance when detecting changes to the system. However, the Final Office Action proffers no further analysis as to why this motivation applies to the problem solved in Gehman in any way.

Furthermore, even given this motivation, the teachings of Grimsrud would not lead a person of ordinary skill in the art to modify the teachings of Gehman in such a way that would result in the presently claimed invention, because neither reference teaches or suggests storing a copy of master data in a second non-legacy computer, executing at least one operation on the copy of the master data, sending the at least one operation to a first non-legacy computer, executing the at least one operation on the master data at the at least one legacy computer, determining whether the first non-legacy computer successfully executed the at least one operation, and synchronizing the master data by applying the at least one operation in response to the first non-legacy computer successfully executed the at least one operation, as recited in claim 1, for example. Rather, a combination of Gehman and Grimsrud would result in a system for

perpetuating changes to master data from an event master server to event client servers where one of the servers may have a peripheral with a legacy failure control mechanism.

The applied references, taken individually or in combination, fail to teach or suggest each and every claim feature. Therefore, Gehman and Grimsrud do not render claim 1 obvious. Because claims 2-5 depend from claim 1, the same distinctions between Gehman and Grimsrud and claim 1 apply for these claims. In addition, claims 2-5 recite additional combinations of features not taught or suggested by the prior art.

Therefore, Appellants respectfully request that the rejection of claims 1-5 under 35 U.S.C. § 103(a) not be sustained.

C. 35 U.S.C. § 103, Alleged Obviousness of Claims 24-31

The Office rejects claims 24-31 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Gehman et al. (U.S. Patent No. 7,136,881) in view of Grimsrud (U.S. Patent No. 6,546,437) and Yousefi'zadeh (U.S. Patent No. 6,950,848). (The Final Office Action states that claims 1-5 are rejected; however, the body of the rejection addresses claims 24-31.) Appellants respectfully traverse this rejection.

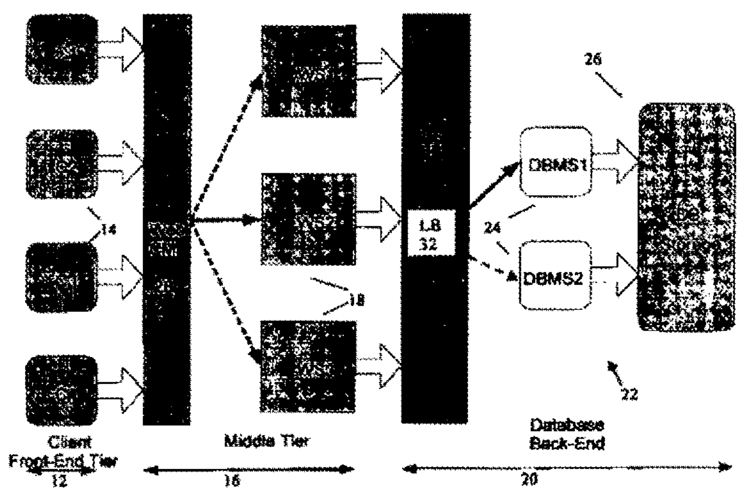
With respect to claim 24, the Final Office Action issued March 10, 2010, states that Yousefi'zadeh teaches receiving, via a first software connector, at least one operation from a thin client computer, wherein the thin client computer stores a copy of master data from a legacy computer and executes the at least one operation on the copy of the master data at col. 6, lines 42-45, which states:

> Further the LB module 32 utilizes at least one interface for connecting to, and communication with, the database servers 24. In one version, the LB module 32 is implemented to transparently communicate with the middle-tier web servers 18 and the back-end tier database servers 24. The LB module 32 is implemented as object-oriented software that can be easily plugged into web browsers 18 and be used for building web-based applications and Graphical User Interfaces (GUIs) along with built in database access methods, security, and networking support. The LB module software can execute on several different computer systems by layering libraries on top of any operating system, and provides a robust method of integrating incompatible computer systems within the framework of e.g. reusable component software, thin client-server software, network programming, secure browser-

based software, threaded multimedia code, etc. For the server side, the LB module software uses servlet technology to make low level details of programming tasks such as raw sockets into a more productive application level, and provides a standard generic and portable mechanism for efficiently developing modular server side applications such as generating dynamic web content.

Yousefi'zadeh, col. 6, lines 29-50. This cited portion of Yousefi'zadeh teaches a load balancing module that can execute on several different computer systems by layering libraries on top of any operating system and provides a method of integrating incompatible computer systems, for example thin client-server software. However, the Final Office Action proffers no analysis or technical reasoning as to why this is somehow equivalent to receiving, via a first software connector, at least one operation from a thin client computer, wherein the thin client computer stores a copy of master data from a legacy computer and executes the at least one operation on the copy of the master data. Yousefi'zadeh is concerned with load balancing requests to multiple database servers; Yousefi'zadeh makes no mention of a thin client storing a copy of master data and executing operations on the copy of the master data.

Still with respect to claim 24, the Final Office Action states that Yousefi'zadeh teaches sequentially replaying the at least one operation on the master data at the legacy computer via a second software connector at FIG. 1A and col. 15, lines 4-25, which are as follows:



10

FIG. 1A

> For each of the above load balancing schemes, the selection of a node at each iteration can be maintained as load balancing history (e.g., in memory or on disk). FIG. 8 shows a functional/data flow diagram of load-balancing data flow in the overall database load-balancing modules 32 in FIG. 3, illustrating decision-making steps for selecting between the two nodes A and B for one or more of each of the above load-balancing schemes 80 (e.g., RR, LNC, RATIO, OBS as weighted combination of LNC and FRT, PRD, etc.). The data flow shows the initialization of the iteration measure k and the status of each node A(k) and B(k) (step 150). Thereafter the iteration measure k is incremented for each iteration (step 152), for each node the LB module 32 receives (k-1) instance choice results from the load history of the node (i.e., candidate/potential node for assignment of query in iteration (k-1)) (step 154), processes the results in different load balancing scheme/steps e.g. RR 156, RATIO 158, LNC 160, FRT 162, PRD 164; and makes the database server choice/selection for iteration k. Said load balancing schemes can be performed sequentially or in parallel, and each of the iterations indicates a measurement factor of the selected node.

This portion of Yousefi'zadeh teaches an architecture of a multi-tier computer system and a functional/data flow diagram of load-balancing data flow in the overall database load-balancing modules. Again, the Final Office Action proffers no analysis or technical reasoning as to why this is somehow equivalent to replaying the at least one operation on the master data at the legacy computer via a second software connector. The Final Office Action appears to cite seemingly arbitrary portions of the reference with no explanation as to their relevance.

Still with respect to claim 24, the Final Office Action acknowledges that Yousefi'zadeh is silent with respect to in response to a determination that the at least one operation is successful, synchronizing the master data by applying the at least one operation to form a modified copy of the master data. The Final Office Action implies that Gehman teaches these features at FIG. 3A, S82, FIG. 3B, S98.

As seen in FIG. 3B of Gehman, step S96 determines whether a sequence number is less than or equal to the stored sequence number. As stated above with respect to claim 1, this determination decides whether to send an event notification to an event client server. Therefore, Gehman teaches a determination concerning the sequence of event notifications, but does not teach a determination of whether a middle-tier computer successfully executes at least one operation on a legacy computer, wherein the at least one operation is received from a thin client

computer. In step S82, Gehman teaches a determination as to whether a modification message exists in a queue to be applied to the master data at the legacy computer. Gehman does not teach that such a modification message corresponds to execution of operation on a **copy** of master data at a thin client computer. Furthermore, the Final Office Action proffers no technical analysis or explanation as to why existence of a modification message in a queue is somehow equivalent to a determination of whether at least one operation received from a non-legacy computer that was executed on a copy of master data was successfully replayed on master data at a legacy computer.

Still further, the Final Office Action acknowledges that Gehman does not teach that one computer is a legacy computer and other computers are non-legacy computers. The Final Office Action alleges that Grimsrud generally teaches both legacy and non-legacy computers and concludes that it would have been obvious to a person of ordinary skill in the art to incorporate legacy and non-legacy computers in Gehman. Appellants respectfully disagree.

Grimsrud teaches communication between a computer and a peripheral with a legacy failure control mechanism. Grimsrud teaches a computer 174 connected to a peripheral 176 via an advanced technology attachment (ATA) interface 178. The computer 174 may send a request for information to the peripheral, and the peripheral may send a reply to the computer that causes the computer to not use the peripheral. See Grimsrud, col. 1, lines 30-56. The computer and peripheral can be any combination of legacy and non-legacy in various embodiments of Grimsrud.

However, Grimsrud does not teach a person of ordinary skill in the art to modify Gehman to include legacy and non-legacy computers. There is no problem in Gehman for which Grimsrud is a solution. The Final Office Action proposes that a person of ordinary skill in the art would have been motivated to combine the teachings of Gehman and Grimsrud "in order to the method of detecting any changes or any hardware connected to the system." While this may be a problem solved in Grimsrud, this motivation does not apply to Gehman in any significant way.

Furthermore, even given this motivation, the teachings of Grimsrud would not lead a person of ordinary skill in the art to modify the teachings of Yousefi'zadeh and Gehman in such a way that would result in the presently claimed invention, because neither reference teaches or suggests storing a copy of master data in a thin client computer, executing at least one operation on the copy of the master data, receiving the at least one operation by a middle tier computer,

sequentially replaying the at least one operation on the master data at the at least one legacy computer, determining whether the middle tier computer successfully executed the at least one operation, and synchronizing the master data by applying the at least one operation in response to the middle tier computer successfully executing the at least one operation, as recited in claim 24, for example. Rather, a combination of Yousefi'zadeh, Gehman, and Grimsrud would result in a system for perpetuating changes to master data from an event master server to event client servers where one of the servers may have a peripheral with a legacy failure control mechanism and a load balancing module load balances requests to a plurality of database servers.

The applied references, taken individually or in combination, fail to teach or suggest each and every claim feature. Therefore, Yousefi'zadeh, Gehman, and Grimsrud do not render claim 24 obvious. Because claims 25-17 depend from claim 24, the same distinctions between Yousefi'zadeh, Gehman, and Grimsrud and claim 24 apply for these claims. Claims 28-31 recite subject matter addressed above with respect to claims 24-27. In addition, claims 25-31 recite additional combinations of features not taught or suggested by the prior art.

Therefore, Appellants respectfully request that the rejection of claims 24-31 under 35 U.S.C. § 103(a) not be sustained.

VIII. Conclusion

In view of the above, Appellants respectfully submit that claims 1-5 and 24-31 of the present application are definite and that the features of these claims are not taught or suggested by the cited prior art. Accordingly, Appellants request that the Board of Patent Appeals and Interferences overturn the rejections set forth in the Final Office Action.

Respectfully submitted,

Stephen R. Tkacs
Reg. No. 46,430
Walder Intellectual Property Law, P.C.
17330 Preston Road, Suite 100B
Dallas, TX 75252
(972) 380-9475
AGENT FOR APPELLANTS

CLAIMS APPENDIX

1. A method of synchronizing data in a distributed data processing system comprising:

storing a master data in at least one legacy computer system;

enabling a first non-legacy computer to support synchronization;

storing a copy of the master data in a second non-legacy computer;

executing, by said second non-legacy computer, at least one operation on said copy of the master data;

sending, by said second non-legacy computer, said at least one operation to said first non-legacy computer;

replaying, by said first non-legacy computer, said at least one operation;

determining whether the at least one operation is successful; and

in response to a determination that the at least one operation is successful, synchronizing said master data by applying said at least one operation to form a modified copy of the master data.

2. The method in claim 1, further comprising sending, by the second non-legacy computer, a synchronization protocol to the first non-legacy computer.

3. The method in claim 1, wherein said at least one operation further comprises at least two operations which are replayed by said first non-legacy computer sequentially.

4. The method in claim 1, wherein the replaying, by said first non-legacy computer further comprises:

sending by said first non-legacy computer the results from said at least one operation, to said second non-legacy computer; and

sending by said first non-legacy computer the modified copy of the master data, to said second non-legacy computer.

5. The method in claim 1, further comprises:

responsive to a determination that the at least one operation is not successful, not synchronizing the master data.

24. An apparatus in a middle-tier computer, comprising:

a processor; and

a memory coupled to the processor, wherein the memory comprises instructions which, when executed by the processor, cause the processor to:

receive, via a first software connector, at least one operation from a thin client computer, wherein the thin client computer stores a copy of master data from a legacy computer and executes the at least one operation on the copy of the master data;

sequentially replay the at least one operation on the master data at the legacy computer via a second software connector;

determine whether the at least one operation is successful; and

in response to a determination that the at least one operation is successful, synchronize the master data by applying the at least one operation via the second software connector to form new master data at the legacy computer, such that in response to a determination that the at least one operation is not successful, the middle-tier computer does not synchronize the master data.

25. The apparatus in claim 24, wherein instructions further cause the processor to:

receive a synchronization protocol from the thin client computer via the first software connector.

26. The apparatus in claim 24, wherein the at least one operation further comprises at least two operations that are replayed by the middle-tier computer sequentially.

27. The apparatus in claim 24, wherein replaying the at least one operation on the master data at the legacy computer further comprises:

sending the results from the at least one operation to the thin client computer via the first software connector; and

sending the new master data to the thin client computer via the first software connector.

28. A computer program product comprising a computer recordable medium having a computer readable program recorded thereon, wherein the computer readable program, when executed on a middle tier computer, causes the middle tier computer to:

receive, via a first software connector, at least one operation from a thin client computer, wherein the thin client computer stores a copy of master data from a legacy computer and executes the at least one operation on the copy of the master data;

sequentially replay the at least one operation on the master data at the legacy computer via a second software connector;

determine whether the at least one operation is successful; and

in response to a determination that the at least one operation is successful, synchronize the master data by applying the at least one operation via the second software connector to form new master data at the legacy computer, such that in response to a determination that the at least one operation is not successful, the middle-tier computer does not synchronize the master data.

29. The computer program product in claim 28, wherein the computer readable program further causes the middle tier computer to:

receive a synchronization protocol from the thin client computer via the first software connector.

30. The computer program product in claim 28, wherein the at least one operation further comprises at least two operations that are replayed by the middle-tier computer sequentially.

31. The computer program product in claim 28, wherein replaying the at least one operation on the master data at the legacy computer further comprises:

sending the results from the at least one operation to the thin client computer via the first software connector; and

sending the new master data to the thin client computer via the first software connector.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE